Zoea

Research Note 23

Coding Isn't a Numbers Game

Edward McDaid & Sarah McDaid
19 Dec 2022

There is a fundamental disconnect between what is regarded as success when building an AI as opposed to code. This has serious implications for AIs that produce code.

Human coders usually produce software with reference to some form of functional requirement - often including a set of acceptance criteria. Before a piece of code is considered releasable it needs to deliver the expected functionality in a sufficiently reliable manner. Someone has given some thought to what the code should do and how this behaviour needs to be verified. The key to successful software development is to recognise the importance of all of these stages - there is more to coding than just coding.

Currently there is a lot of interest in getting AIs to produce software. Most of the systems produced recently were built by training a model on lots of examples of other people's code. We won't say anything here about the ethical questions raised by this approach. In any event this does two things. First it 'teaches' the model the syntax of whatever programming language is being used. The model also 'learns' something of the semantics of many different programming domains and problems. So if you ask such an AI for code to calculate the area of a circle you're probably in luck. This seems great.

Unfortunately most of the people who currently build AIs have a different interpretation of 'success' than do coders. Say you're building an AI to recognise cats using a deep learning approach. After you've trained your model on lots of pictures of cats, the yardstick for success is what percentage of test pictures are recognised correctly. AI practitioners seem to be pretty pleased if they achieve 80 or 90 percent. That may well be good enough for

cat recognition, for most people - after all recognising cats isn't usually a matter of life or death. Except similar AIs are also being used for more serious problems like moderating social media, deciding on a loan, and in the not too distant future writing code.

When it comes to code: coders, end users and the general public see success in different terms. Either a requirement has been met or it hasn't. Having a feature that works say 97% of the time is regarded as a bug. Increasing the percentage doesn't really help much either - even in areas that are non-life threatening - it just makes the bug more pernicious. No-one expects their bank balance to be wrong or their email to be misdirected any percentage of the time. As far as code is concerned success is binary

Fortunately this mismatch doesn't apply to all AI approaches. Zoea is an AI that is built through coding. It encodes explicit programming language and software development knowledge. As a result it completely avoids the need for any sort of training and it doesn't use a single line of anyone else's code. Users input explicit functional requirements in the form of test cases so the software it produces is always guaranteed to meet the requirements. Deep learning and related approaches are a great way to build some sorts of AI. However AI that produces code needs firmer foundations.

Learn more at **zoea.co.uk**